

*Citation for published version:*

Huang, B, Bryson, JJ & Inamura, T 2013, Learning motion primitives of object manipulation using Mimesis Model. in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, pp. 1144-1150, 2013 IEEE International Conference on Robotics and Biomimetics, ROBIO 2013, Shenzhen, China, 12/12/13. <https://doi.org/10.1109/ROBIO.2013.6739618>

*DOI:*

[10.1109/ROBIO.2013.6739618](https://doi.org/10.1109/ROBIO.2013.6739618)

*Publication date:*

2013

*Document Version*

Early version, also known as pre-print

[Link to publication](#)

**University of Bath**

## **Alternative formats**

If you require this document in an alternative format, please contact:  
[openaccess@bath.ac.uk](mailto:openaccess@bath.ac.uk)

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

### **Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Learning Motion Primitives of Object Manipulation Using Mimesis Model

Bidan Huang<sup>1</sup>, Joanna Bryson<sup>1</sup> and Tetsunari Inamura<sup>2</sup>

**Abstract**—In this paper, we present a system to learn manipulation motion primitives from human demonstration. This system, based on the statistical model “Mimesis Model”, provides an easy-to-use human-interface for learning manipulation motion primitives, as well as a natural language interface allowing human to modify and instruct robot motions. The human-demonstrated manipulation motion primitives are initially encoded by Hidden Markov Models (HMM). The models are then projected to a topological space where they are labeled, and their similarities are represented as their distances in the space. We then explore the unknown area in this space by interpolation between known models. New motion primitives are thus generated from the unknown area to meet the new manipulation scenarios. We demonstrate this system by learning bimanual grasping strategies. The implemented system successfully reproduces and generalizes the motion primitives in different grasping scenarios.

## I. INTRODUCTION

Objects and tool manipulation is a key skill for service robots. Manipulation is a sequence of motions that changes the object’s status. A high dimensional search space makes this sequence of motion difficult to generate. To reduce the search space, the concept of motion primitives has been introduced to robot planning from neuroscience [1]. The basic principle is to discretise a manipulation task into a set of motion primitives, each of which serves as an elementary manipulation function. After modeling each primitive, the whole task then can be achieved by coordinating them using an action selection system.

Modeling motion primitives remains an open problem. Many articles discuss how to design motion primitives that accomplish specific tasks [15], [5], [9]. In these works, motion primitives are modeled as a set of differential equations or control rules. New motions are generated by tuning the parameters in the models.

In this paper we propose a system for learning manipulation motion primitives from human demonstration [13], [4]. To achieve this goal, we exploit our previous work on the Mimesis Model [11]. The Mimesis Model is a mathematical realization of the function of the mirror neurons, which fire both when an animal observes and when it executes a motion. Neuroscience research shows connections between the mirror neurons and an animal’s imitation learning mechanism [18]. The mimesis model has been shown to be effective in motion

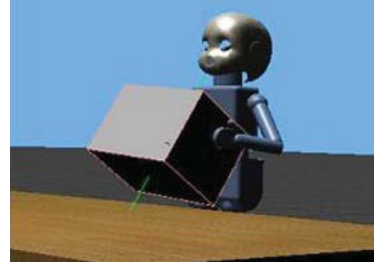


Fig. 1: iCub grasping a box by both arms

recognition, generation and robot imitation learning [10], [16]. While our previous work has focused on learning whole body movement, the present paper extends the Mimesis Model to learn motions of manipulation that involve interaction with objects.

In the Mimesis Model, all demonstrated motion patterns are projected to a topological space, called “proto-symbol space”. In this space, the similarity between two motions are represented as the Euclidean distance between their projected points. Recognition of an unknown motion is done by finding its closest known motion, while generating a new motion is done by interpolation between the new motions. Further, the motions are symbolized in this space by human labeling of their projection points with the manipulation’s effects, such as “grasp low box” or “grasp high box”. As a result, this system provides a natural language interface allowing the adjustment of the robot’s motion. For example, starting from the “grasp low box” motion, we can instruct the robot to raise its arms higher to grasp a box on the top of a cabinet by the command “not high enough, go higher to grasp”. This command could result in generating a motion closer to the motion labeled by “grasp high box”.

The work of Kunori et al. [14] using hidden Markov models to encode motion primitives for object manipulation shares a similar concept to our work. While they focused on extracting key features and reshaping movements for good performance, we focus on combining known manipulation motion primitives to generate new motions that can achieve the desired effects. Although interpolation of known motions is not new in motion synthesis [7], [6], most of the existing work focus on free body motion. The application to object manipulation is rarely discussed.

The goal of this work is to enable robot to learn manipulation motion primitives and generate new motions to adapt to unseen scenarios. The system is implemented for a bimanual grasping task. Unlike static fingertip grasping synthesis [8], in this task we focus on the grasp reaching motion.

<sup>1</sup>Bidan Huang (b.huang@bath.ac.uk) and Joanna Bryson (j.j.bryson@bath.ac.uk) are with Intelligent Systems Group (IS), Computer Science Department, University of Bath, UK

<sup>2</sup>Tetsunari Inamura (inamura@nii.ac.jp) is with the National Institute of Informatics / Dept. of Informatics, The Graduate University for Advanced Studies, Japan

This paper is organized as follows. Section II details the proposed method: section II-A illustrates how we demonstrate the grasping strategies; section II-B explains the concept of proto-symbol space and how to create it; section II-C details the learning process of the physical meaning of the proto-symbols and II-D explains how to generate new grasping motions for unseen objects. Section III shows the experimental results, followed by the conclusion and discussion in Section IV.

## II. METHODOLOGY

We adopt the Mimesis Model to learn motion primitives of manipulation from human demonstrations. In this approach, the demonstrated motions are firstly encoded by Hidden Markov Models (HMM). A topological space, called the proto-symbol space, is constructed using gaussians to represent the similarities between the primitives within the HMM and allow labelling. In this space, each primitive is abstracted to a point (proto symbol). New motions are generated by locating new points. The correlation between the location of the new points and their physical effects is learned by regression. This correlation allows us to directly query a new motion by a high level task requirement.

In short, the general approach has 5 steps and is described as follows:

- 1) **Demonstration:** A human teacher demonstrates manipulation motion primitives (section II-A).
- 2) **Abstraction:** Abstract the motion primitives by HMM and create the proto-symbol space (section II-B).
- 3) **Interpolation:** Interpolate the proto-symbol space and construct new HMM. (section II-C).
- 4) **Generating:** Generate motion using proto symbols (section II-D).
- 5) **Learning:** The robot reproduces the motion and learns the correlation between the location of the proto symbols and the effect of the generated motions (section II-E).

### A. Demonstration

The motion primitives of manipulation are first demonstrated by a human. The same primitives are demonstrated a few time so that the HMM is able to encode the general features of the movement. Each primitive corresponds to the movement in one scenario. To enable the robot to work in different scenarios, different primitives need to be demonstrated. For example, to design a motion primitive for fetching boxes in different sizes, we need to demonstrate at least two primitives: grasping a small box and grasping a big box (Figure 2). Grasping a box with the size between the big one and the small one may then be achieved by interpolation between these primitives. For more complex motion, more than two primitives may be required.

In this approach, the demonstrated motions provide not only the dynamics of the motion primitives, but also define the feasibility of the motions. As the new motions are interpolations of the demonstrations, joint limits or singularities can be avoided by well-defined demonstrations.

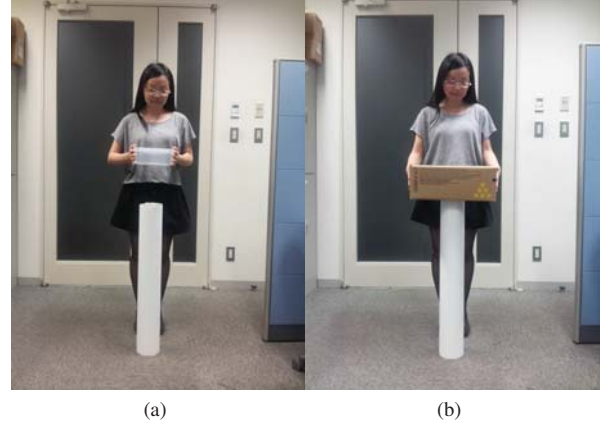


Fig. 2: Human bi-manual grasps. (a) A human grasping a small box. (b) A human grasping a big box

### B. Abstraction

In this step, the demonstrated motion primitives are abstracted to “proto symbols” in the proto-symbol space, where the similarity between different motion patterns are represented as Euclidean distance. The abstraction is done by encoding the motion patterns in Hidden Markov Model (HMM).

An HMM is a stochastic mathematical framework for learning sequential data. It describes the stochastic sequences as Markov chains, where the states of the sequences depends only on the previous state. In this work we use a left-to-right continuous Hidden Markov Model (CHMM) to encode the motion primitives (Figure 3).

Each motion pattern is described by a set of variables:  $\lambda = \{Q, A, B, \pi\}$ , where  $Q = \{q_1, \dots, q_N\}$  is a finite set of states,  $A = \{a_{ij}\}$  is the state transition probability matrix denoting the probability that node  $q_i$  transits to  $q_j$ ,  $B = \{b_i\}$  is the continuous output probabilities denoting the probability distribution that the output vector  $o[t]$  is given by  $q_i$ , and  $\pi$  is the initial distribution. The  $\pi$  is the same as for each CHMM as we use a left-to-right CHMM model. Therefore, the parameter set  $\mathcal{P} = \{a_{ij}, b_i\}$  characterizes the behavior of a stochastic process. We call  $\mathcal{P}$  a proto symbol.

The CHMM is learned using the Baum-Welch algorithm. For simplicity, we use a single Gaussian model for the output of each node in the CHMM. This allows us to synthesis new motions simply by interpolating the means and covariances of the Gaussians (see section II-C).

The proto-symbol space is constructed to represent the similarity between CHMMs. This requires us to compute the similarity between each pair of CHMMs. In this work, we use the Bhattacharyya distance [12] as our similarity metric, as it is a symmetric metric with respect to two probability variables. The Bhattacharyya distance  $BD(p, q)$  between two Gaussian distributions  $p(x; \mu_p, \Sigma_p)$  and  $q(x; \mu_q, \Sigma_q)$  is defined as follows:

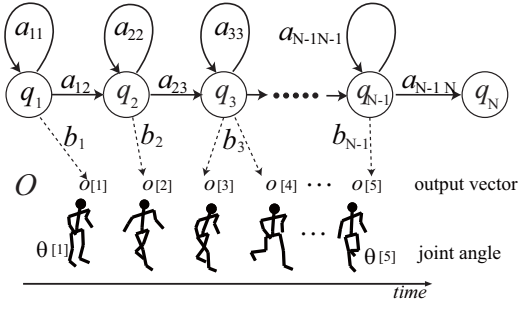


Fig. 3: An illustration of encoding a motion by Continuous Hidden Markov Model

$$BD(p, q) = -\log \int_{-\infty}^{\infty} \sqrt{p(x)q(x)} dx$$

$$= \frac{1}{8} \mu_{pq} \left( \frac{\Sigma_p + \Sigma_q}{2} \right)^{-1} \mu_{pq}^T + \frac{1}{2} \log \frac{|\frac{\Sigma_p + \Sigma_q}{2}|}{|\Sigma_p|^{\frac{1}{2}} |\Sigma_q|^{\frac{1}{2}}} \quad (1)$$

where

$$\mu_{pq} = \mu_p - \mu_q \quad (2)$$

The Bhattacharyya distance  $DB(\lambda_1, \lambda_2)$  between two HMMs is computed by summing the distances between the Gaussian distributions, i.e. the output probability distributions for the nodes:

$$DB(\lambda_1, \lambda_2) = \sum_i \sqrt{BD(\mathcal{N}_{1i}(\mu_{1i}, \Sigma_{1i}), \mathcal{N}_{2i}(\mu_{2i}, \Sigma_{2i}))} \quad (3)$$

where  $\mathcal{N}_{ji}(\mu_{ji}, \Sigma_{ji})$  is the output probability at the  $i$ -th node  $q_i$  of the HMM  $\lambda_i$ .

The proto-symbol space is constructed using the multi-dimensional scaling technique (MDS) [19]. This technique computes the locations of the CHMMs in the proto-symbol space by minimizing the criterion:

$$S^2 = \sum_{i,j} (DB_{ij} - d_{ij}) \quad (4)$$

where  $DB_{ij}$  is the Bhattacharyya distance between the  $i$ th and  $j$ th CHMMs and  $d_{ij}$  is their Euclidean distance between their proto symbols. Figure 4 shows a proto-symbol space constructed by 4 proto symbols.

### C. Interpolation

To generate a new motion, a new location in the proto-symbol space is exploited. This is done by interpolation between different proto symbols. In the left-to-right model, the expected duration  $s_i$  of the state  $q_i$  can be computed as

$$s_i = \sum_{n=1}^{\infty} n(1 - a_{ii})a_{ii}^{n-1} = \frac{1}{1 - a_{ii}}, \quad (5)$$

where  $a_{ii}$  is the probability of self-transition at the state  $q_i$ .

A new proto symbol  $\hat{P}$  is expressed by the linear combination of  $m$  proto symbols  $(\mathcal{P}_1, \dots, \mathcal{P}_m)$ . The weights of different proto symbols are expressed by the mix coefficient

$c_j$ . The expected duration  $\hat{s}_i$  for the new motion in the state  $q_i$  is computed as

$$\hat{s}_i = \sum_j^m c_j s_i^{(j)} \quad (6)$$

with this we can compute the new state transition probability  $\hat{a}_{ii}$  as

$$\hat{a}_{ii} = \frac{\hat{s}_i - 1}{\hat{s}_i} \quad (7)$$

Note that according to Eq. 6,  $s_i \geq 1$  and hence the following constraint must be satisfied

$$\sum_j^m \frac{c_j}{1 - a_{ii}^{(j)}} \geq 1 \quad (8)$$

To compute the new output probability  $b_i$ , since there is only one Gaussian in each state, we simply sum the means and variances of the Gaussians in the same state of different HMMs as

$$\hat{b}_i(O) = \mathcal{N}(O; \hat{\mu}_i, \hat{\sigma}_i^2) = \sum_j^m c_j b_i^{(j)}(O) \quad (9)$$

where

$$\hat{\mu}_i = \sum_j^m c_j \mu_i^{(j)} \quad (10)$$

$$\hat{\sigma}_i^2 = \sum_j^m c_j^2 \sigma_i^{(j)2} \quad (11)$$

$\mu_i^{(j)}$  and  $\sigma_i^{(j)}$  are the mean and variance of the Gaussian representing the  $i$ -th state.

In theory this method can also be used to extrapolate the proto symbols with a negative mixing coefficient, which allows us to explore outside the feasible region defined by the demonstrations. This could generate motions outside of the robot's experience. However the feasibility can not be guaranteed, e.g. this may give joint angles over the robot's limit.

### D. Generating

A new motion sequence is generated from the new proto-symbols by using an averaging method [11]. The steps for generation are as follow :

- 1) Starting from a node  $q_1$ , let the motion element sequence be  $O = \phi$ .
- 2) Using the transition probability  $\{a_{ij}\}$  to generate the states  $q_j$ .
- 3) Using the output probabilities  $\{b_i\}$  to decide the output label  $o_k$ .
- 4) Adding the output label  $o_k$  to the motion elements sequence  $O$ .
- 5) Stop when the generation process reaches the end node  $q_N$ .

Due to the stochastic nature of this method, motions generated by the same HMM are not identical at each time.

Nevertheless, they have the same dynamics as they are generated from the same parameters  $A$  and  $B$ . We repeat the above steps and average the generated motions to produce the final motion. As the duration of each generated motion is different, before averaging we uniform the time in each motion by:

$$\bar{\theta}(t) = \theta\left(T \frac{t}{T_u}\right) \quad (12)$$

where  $T$  is the time duration of each motion, and  $T_u$  is the uniformed time duration. After this joint angles are averaged amount all generated motions.

### E. Learning

Unlike free body motions, motion for object manipulation needs to achieve certain outcome, such as grasping a given size of box. However the physical effects of the demonstrated and generated motions are unknown, as the robot have a different embodiment from the human demonstrator. For example, the motion for a human to grasp a 30cm length box may only allow the robot to grasp a 15cm length box. Therefore, a learning process is needed to quantify the correlation between the location of the new proto symbol and its physical effect.

To do this, we first interpolate the proto-symbol space with a few different mixing coefficients. We then generate the corresponding motions and perform them with a robot. The platform we used is the iCub in the Webots simulator. As the iCub has the same joint configuration of arm as the one provided by Kinect, we directly apply the generated motions to the iCub. The outcome of the motion, for example the size of the box the robot can grasp with the motion, are recorded with their corresponding mixing coefficients.

The correlation between the sizes and the mixing coefficients is then found by regression analysis. Figure 6 shows an example of the result of the regression. With this result, we are able to infer the mixing coefficient for generating a proper motion. By using the method detailed in section II-D, the motion with a desired effect can be generated. Our experiments verify that this method can generate new grasping motions and the result will be discussed in the next section in details.

## III. EXPERIMENT

This section presents the implementation of the system in learning bi-manual grasping motion primitives. Bi-manual grasping is regularly used in daily life. One of the most commonly used strategies is putting two hands at the opposite sides of a bulky object to apply antipodal grasps (Figure 2). The motion primitives, including an approaching motion and a lifting motion, can be used to grasp many different objects. In our experiment, we focus on learning this strategy and verify that it can be generalized to grasp objects in unseen scenarios.

The strategy is demonstrated in two different scenarios: grasping boxes with different sizes and grasping boxes placed on different heights. As explained in section II-A, the

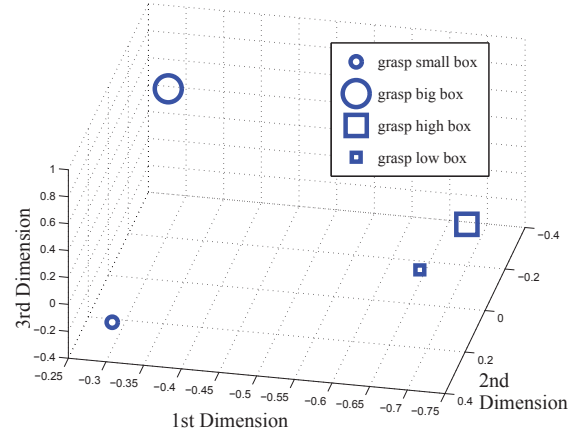


Fig. 4: Proto-symbol space constructed by four motion primitives

demonstrations are chosen to define the boundary situations of the grasping motions. In this experiment, four different motion primitives are demonstrated: grasping the biggest feasible box, grasping the smallest feasible box, grasping the lowest feasible box and grasping the highest lowest box. Objects with size or height outside the feasible area might be able to be grasped by the same strategy, but the motion may be very close to infeasible joint angles, or not be natural for human behavior. In our case, the bi-manual grasp of a box longer than the distance between the left and right elbow is very difficult for the iCub; bi-manual grasp of a very small size box is possible but human would normally use a single-hand grasp.

All the demonstrated motion sequences are recorded by Kinect, a skeleton tracking device widely used both in gaming industry and academic research [17]. It is a markerless stereo camera which can automatically detect and track human joint configuration. The output data from Kinect is converted to joint angle space.

In this experiment, the grasping motions only involve the arms. The objects are placed in the working space of the human demonstrator so that the human does not need to change the location to grasp the objects. Due to the technical limitations of Kinect, it can not record the wrist joint and hence wrist is omitted in our current experiment. In total, 8 degrees of freedom are recorded in the human motion: left shoulder (3D), right shoulder (3D), left elbow (1D) and right elbow (1D). When the hands contact the objects, the wrist joints will change due to the force applied by the arm. This adds extra uncertainties to the grasping motion, as well as a certain amount of compliance. As a result the box may rotate some angle after lifting (Figure 10).

Each grasping motion is demonstrated five times. In all demonstrations, the starting postures are the starting posture used by Kinect: the  $\Psi$  pose that with two arms raising over the head, both palms facing inside.

The raw data are noisy due to the limitation of the motion capture device. To denoise the motion signal, we used second-order low-pass filters to smooth the motion outputs



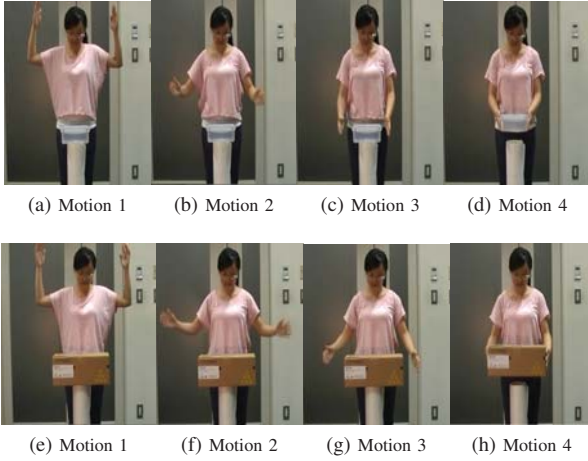


Fig. 5: (a)-(d): Human demonstrating bi-manual grasp of a small box (size  $20\text{cm}(\text{length}) \times 15\text{cm}(\text{width}) \times 10\text{cm}(\text{height})$ ). (e)-(h): Human demonstrating bi-manual grasp of a big box (size  $40\text{cm}(\text{length}) \times 20\text{cm}(\text{width}) \times 15\text{cm}(\text{height})$ )

and remove high frequency noise caused by vibration of the machinery. Each motion is low-pass filtered by 1Hz, 5Hz and 10Hz and all the filtered results are supplied as the training data for the Mimesis Model. The demonstrated motions are performed by the Webots iCub to find out their outcomes.

In the abstraction step (section II-B), the four motion primitives are encoded by four CHMMs. To completely distinguish between four points we need at least a three dimensional space. Hence we construct a three dimensional proto-symbol space by using the MDS with these CHMMs (Figure 4). To generate new grasping motions, we interpolate (section II-C) the proto-symbol space with different mixing coefficients. New motions are then generated at each of the interpolation points as detailed in the section II-D. These generated motion are then performed by the Webots iCub to examine their effects.

All motions are modeled in ten states, determined by five-fold cross validation, and each state is represented by a single Gaussian to keep the simplicity.

#### A. Grasping different sizes boxes

In this scenario we demonstrate the strategies of grasping different sizes of boxes. The boxes are placed on a cylindrical stand with  $84\text{cm}$  height. The human demonstrator stands  $20\text{cm}$  in front of the cylindrical stand (Figure 5).

Figure 5 shows the motion sequences. As can be seen from the figure, for grasping the small box, the hands move directly to it, while for grasping the big box, the arms first open to create a certain distance between hands and then close to reduce the distance until contact with the box. This is to avoid unwanted collision with the box during reaching.

New motions are then generated by mixing the demonstrations. To learn the effect of the motions, all demonstrated and generated motions are performed by the robot. The sizes of the boxes are initially estimated by forward kinematics, and then verified by robot executing the motion to grasp a box. The motion that can hold a box and lift it vertically without any slipping is considered to be a successful grasp.

Mixing Coefficient of the Interpolation Points	Box size of successful grasps (cm)
0(small box) 1(big box)	43
0.2(small box) 0.8(big box)	39
0.5(small box) 0.5(big box)	35
0.8(small box) 0.2(big box)	28
1(small box) 0(big box)	25

TABLE I: Mixing coefficient of the interpolation points and the box size of successful grasps (training)

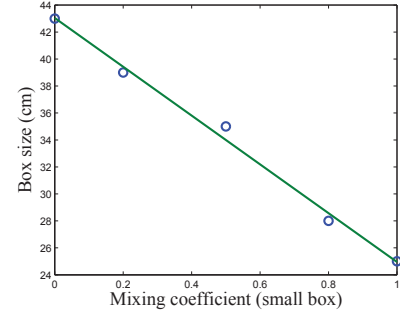


Fig. 6: Linear regression of the interpolation points

Table I shows the mixing coefficients of the motions and the corresponding size of boxes of successful grasps. Note that mixing coefficients always sum to 1. When we make the mixing coefficient to be 1 for one motion and 0 for the other, the generated motion simply corresponds to the motion with mixing coefficient 1. Linear regression is then applied to find out the correlation between the mixing coefficients and the box sizes.

Figure 6 shows the linear regression result of the mixing coefficients and the size of successfully grasped boxes. With the regression results, given a size of box, the mixing coefficient of generating a corresponding grasping motion can be deduced. To test this method, we apply this method to grasp four un-demonstrated boxes with different sizes. All of them can be successfully lifted by the synthesis grasping motions. Table II lists the given boxes size and the computed mixing coefficient and Figure 10 shows the corresponding motions.

#### B. Grasping boxes from different positions

In this scenario the goal is to grasp boxes from different heights. Two motions are demonstrated to grasp a high and a low box. In the demonstrations the high box is placed at the height of  $150\text{cm}$  and the low box is placed at  $70\text{cm}$ . In this case, the two demonstrations are not only different in the arm trajectories but also different in the time duration

Given Box Size	Predicted Mixing Coefficient
27	0.89(small box) 0.11(big box)
30	0.72(small box) 0.27(big box)
36	0.44(small box) 0.56(big box)
40	0.16(small box) 0.74(big box)

TABLE II: Given Box Sizes (cm) and the Predicted Mixing Coefficient (testing)

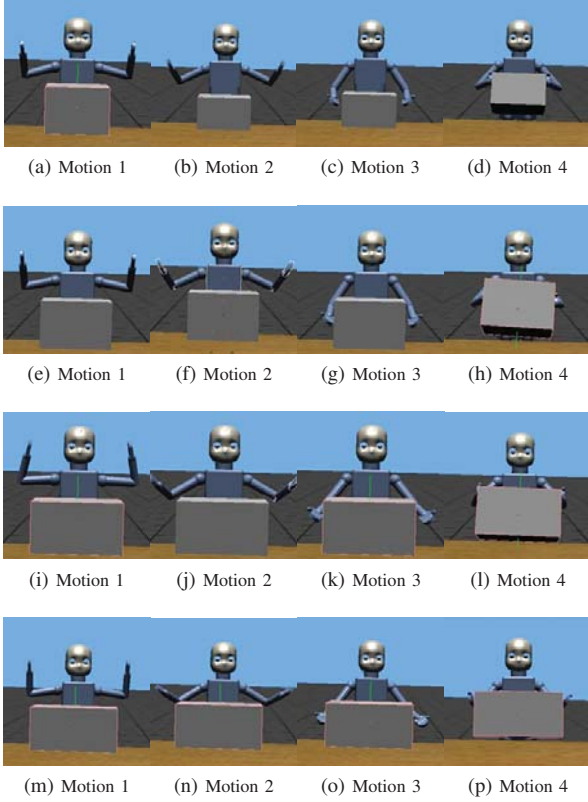


Fig. 7: Robot grasping different boxes with the generated motions. (a)-(d) Box size 0.27cm. (e)-(h) Box size 0.3cm. (i)-(l) Box size 0.35cm. (m)-(p) Box size 0.4cm.

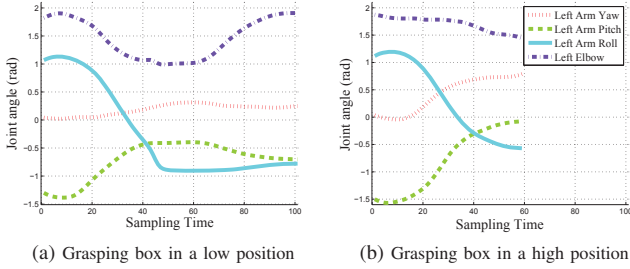


Fig. 8: (a) Left arm motion of a human demonstration of grasping a low box. (b) Left arm motion of a human demonstration of grasping a high box.

(Figure 8). The motion of grasping the high box lasts shorter than grasping the low box as the initial hand position is closer to the box position. At the same time the lifting parts of the motions are different: for the high box the lifting distance is smaller than the low box because of the joint limits of the arms.

Following the same process as described above, we interpolate between the motions for grasping a low box and a high box (Table III). We apply linear regression and hence find out the correlation between the mixing coefficients and the heights of the box (Figure 9).

With the learned correlation, we query the mixing coefficients for four different un-demonstrated heights (Table IV). The generated motions are tested with the Webots iCub, which successfully lifted all the boxes. Besides the height of the box, the time of performing the task and the lifting

Mixing Coefficient	Box height(cm)
0(high box) 1(low box)	49
0.2(high box) 0.8(low box)	53
0.5(high box) 0.5(low box)	58
0.8(high box) 0.2(low box)	62
1(high box) 0(low box)	64

TABLE III: Mixing coefficients of the interpolation points and the box heights (center of mass from the ground) of successful grasps (training).

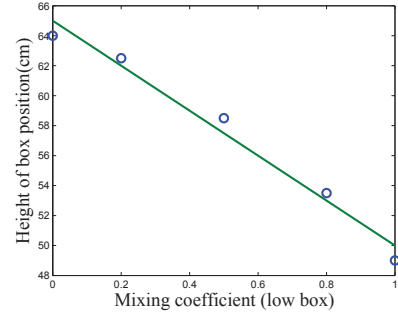


Fig. 9: Linear regression of the interpolation points

distances are also interpolated.

#### IV. CONCLUSIONS

The system presented in this paper uses the Mimesis Model to learn motion primitives for object manipulation. It provides an easy-to-use interface for robot motion generation. Motion primitives are the elementary motions that accomplish basic functions. Neuroscience also indicates the value of reducing the degrees of freedom, since the vertebrate motor system seems to generate motions by combining a small number of motor primitives [1], [3]. Our experiment shows that by combining two motion primitives (8 d.o.f) we can indeed generate many different motion patterns. This framework substantially simplifies the modeling of motion primitives and therefore of control more generally.

In our system, each motion primitive is symbolized in the proto-symbol space and labeled by its effects, e.g. “grasp high box”, “grasp low box” and etc. This provides a linguistic interface for the human to instruct robot motion, by giving language instructions like “go lower to grasp the box”. These primitives can be interpreted as behavior modules, allowing integration with standard action-selection systems for modular AI. Putting more symbolized motion primitives in the proto-symbol space allows us to give more complex instruction and the robot to generate more complex motions. Further, new instructions might be learned autonomously

Given Box Height	Predicted Mixing Coefficient
50	0.02(high box) 0.98(low box)
55	0.34(high box) 0.66(low box)
60	0.66(high box) 0.34(low box)
63	0.86(high box) 0.14(low box)

TABLE IV: Given Box Height (cm) and the Predicted Mixing Coefficient (testing)

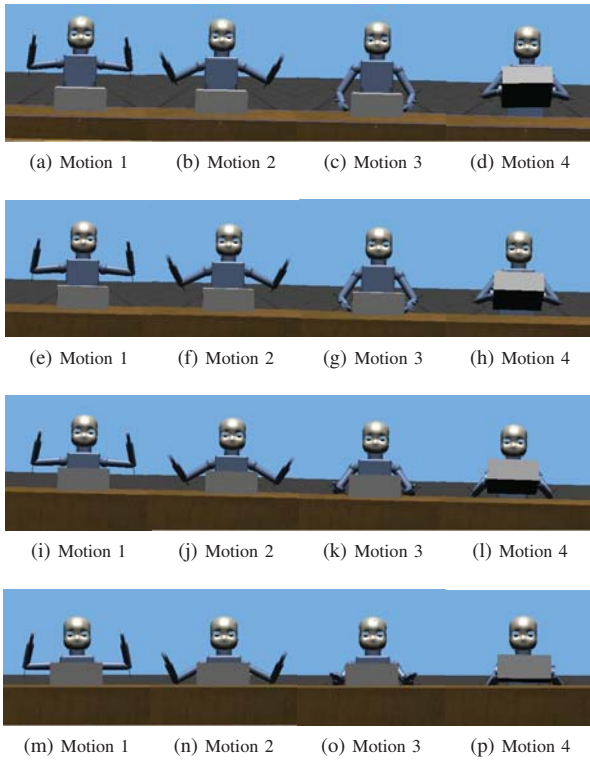


Fig. 10: A robot grasping boxes from different heights with generated motions. (a)-(d) Box at height 50cm. (e)-(h) Box at height 55cm. (i)-(l) Box at height 60cm. (m)-(p) Box at height 63cm.

by combining the observation of new terms with imitation learning [2]. Due to the low-dimensional projection, the robot should be able to adjust the mixing coefficient and generate new motions to execute a command. Despite the human and robot having different embodiments, the proto-symbol space allows the robot and human to communicate.

We implement this system in the Webots simulator with the iCub robot. Two different sets of primitives are learned for the bimanual grasp: grasping different sizes of boxes and grasping boxes from different locations. Interpolation in the proto-symbols space produces new motion primitives that enable the robot to successfully grasp boxes in different scenarios. The correlation between the new motions and their effects are learned using first order linear regression. In our future work of learning more complex motion primitives, higher order or nonlinear regression may need to be employed.

The presented experiment provides a good starting point for our future study in learning more motion primitives for object manipulation. It shows that it is possible to directly adjust parameters in the operational space, without fine tuning variables in the model. This has the advantage, from the user point of view, of planning the motion primitives more intuitively. In this paper, we show that with the Mimesis Model control in one dimension (object size, object height) works effectively. In the future work, we will further study the control in multiple dimensions, through interpolation

between multiple proto symbols.

## REFERENCES

- [1] E Bizzi, VCK Cheung, A d'Avella, P Saltiel, and Me Tresch. Combining modules for movement. *Brain Research Reviews*, 57(1):125–133, 2008.
- [2] Joanna J. Bryson. Embodiment versus memetics. *Mind & Society*, 7(1):77–94, June 2008.
- [3] Joanna J. Bryson. Structuring intelligence: The role of hierarchy, modularity and learning in generating intelligent behaviour. In David McFarland, Keith Stenning, and Margaret McGonigle, editors, *The Complex Mind: An Interdisciplinary Approach*, pages 126–143. Palgrave-Macmillan, Basingstoke, June 2012.
- [4] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):286–298, 2007.
- [5] Javier Felip, Janne Laaksonen, Antonio Morales, and Ville Kyrki. Manipulation primitives: A paradigm for abstraction and execution of grasping and manipulation tasks. *Robotics and Autonomous Systems*, 2012.
- [6] Pascal Glardon, Ronan Boulic, and Daniel Thalmann. Pca-based walking engine using motion capture data. In *Computer Graphics International, 2004. Proceedings*, pages 292–298. IEEE, 2004.
- [7] Kiyoshi Hoshino. Interpolation and extrapolation of repeated motions obtained with magnetic motion capture. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 87(9):2401–2407, 2004.
- [8] Bidan Huang, Sahar El-Khoury, Miao Li, Joanna J. Bryson, and Aude Billard. Learning a real time grasping strategy. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 593–600, 2013.
- [9] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.
- [10] Tetsunari Inamura and Tomohiro Shibata. Geometric proto-symbol manipulation towards language-based motion pattern synthesis and recognition. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 334–339. IEEE, 2008.
- [11] Tetsunari Inamura, Iwaki Toshima, Hiroaki Tanie, and Yoshihiko Nakamura. Embodied symbol emergence based on mimesis theory. *The International Journal of Robotics Research*, 23(4-5):363–377, 2004.
- [12] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *Communication Technology, IEEE Transactions on*, 15(1):52–60, 1967.
- [13] Dana Kulić, Christian Ott, Dongheui Lee, Junichi Ishikawa, and Yoshihiko Nakamura. Incremental learning of full body motion primitives and their sequencing through human motion observation. *The International Journal of Robotics Research*, 31(3):330–345, 2012.
- [14] Hiroto Kunori, Dongheui Lee, and Yoshihiko Nakamura. Associating and reshaping of whole body motions for object manipulation. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5240–5247. IEEE, 2009.
- [15] Paul Michelman and Peter Allen. Forming complex dextrous manipulations from task primitives. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 3383–3388. IEEE, 1994.
- [16] Keisuke Okuno and Tetsunari Inamura. Motion coaching with emphatic motions and adverbial expressions for human beings by robotic system-method for controlling motions and expressions with sole parameter. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3381–3386. IEEE, 2011.
- [17] Gang Ren and Eamonn O'Neill. 3d marking menu selection with free-hand gestures. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 61–68. IEEE, 2012.
- [18] Giacomo Rizzolatti and Laila Craighero. The mirror-neuron system. *Annu. Rev. Neurosci.*, 27:169–192, 2004.
- [19] Susan S Schiffman, M Lance Reynolds, and Forrest W Young. *Introduction to multidimensional scaling: Theory, methods, and applications*. Academic Press New York, 1981.